2022 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2022, Virtual, December 12-15, 2022

Analysis of dynamical systems using symbolic regression

Soichiro Kanaya [1], Toma Takano[1], Satoshi Sunada[2,3] and Tomoaki Niiyama[2]

[1]Graduate School of Natural Science and Technology, Kanazawa University
Kakuma-machi Kanazawa, Ishikawa 920-1192, Japan
[2]Faculty of Mechanical Engineering, Institute of Science and Engineering,
Kanazawa University Kakuma-machi Kanazawa, Ishikawa 920-1192, Japan
[3]Japan Science and Technology Agency (JST), PRESTO,
4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan

Email: kanaya_s@stu.kanazawa-u.ac.jp

Abstract

We study a symbolic regression technique to infer the equations of systems from the observed numerical data. Our method is based on the AI-Feynman, proposed by Udrescu et al., which uses neural networks to detect features of the data, and genetic programming, which is an efficient formula search method that mimics biological evolution. In this study, we show that our method can successfully infer simple equations from measurement data with the aid of the AI-Feynman.

1. Introduction

Behind a seemingly complex phenomenon may lie a simple mathematical formula. However, the method for discovering such equations is not simple. It is one of the most important challenges for physicists to find equations that adequately describe a phenomenon from observational data alone. Recent advances in machine learning technology have made it possible to numerically map observed data onto inferred data, however, the input-output relationship is described as a black box and is difficult for humans to understand. If the relationship can be expressed in a simple form that is understandable for human beings, it will not only advance our understanding of the subject, but also greatly reduce the number of models describing the phenomenon. Symbolic regression has recently been attracting considerable attention as an approach to equation discovery, and commercial software, such as Eureqa and Turing Bot, have been developed [1,2]. Symbolic regression has the potential to be used not only for analyzing actual phenomena, but also for understanding black boxes such as neural network (NN) and improving simulation speed, since it has the potential to convert input-output relationships into simple symbolic representations that are easy to interpret and computationally inexpensive [3].

ORCID IDs Soichiro Kanaya:0000-0003-1475-8755,
Toma Takano:0000-0003-2179-9803,
Satoshi Sunada:0000-0003-0466-8529,
Tomoaki Niyama:0000-0003-3808-4839

Recently, AI-Feynman was proposed by Udrescu et al. as a method for symbolic regression [4,5]. This method is expected to discover laws of separability and symmetry from a trainig dataset consisting of input and output data pairs, and to find equations efficiently. However, this method uses Brute Force as the final search method, which is less efficient than genetic programming, which has been used as a method for inferring equations from data. Therefore, in this study, we combine the two techniques , genetic programming and AI-Feynman. We show that by pre-processing data with the aid of AI-Feynman's method, a simple equation can be well inferred from observed numerical data.

2. Method

As shown in Fig. 1, our proposed method consists of two steps: preprocessing of the observed data and an algorithm for formula discovery. The former is done using the AI-Feynman and the latter using genetic programming.

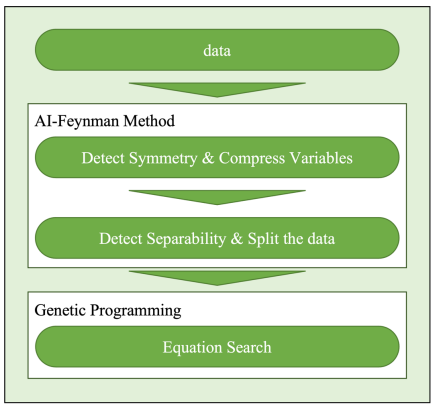


Fig. 1. The Algorithm Flow.

2.1. Law detection and data conversion

AI-Feynman is used to detect the laws of symmetry and separability in the input-output data pairs and to create new data for which equations can be easily obtained by symbolic regression. Symmetry and Separability are expressed by Eq. (1) and Eq. (2), respectively.

$$F(x_0, x_1, \ldots, x_M) = F(x_0 + x_1, x_2, \ldots, x_M) \quad (1)$$
$$F(\boldsymbol{x_A}, \boldsymbol{x_B}) = g(\boldsymbol{x_A}) + h(\boldsymbol{x_B}) \quad (2)$$

Where $x_i \in \mathbb{R}$ is the input value for i-th index, $F(\boldsymbol{x})$ is the output value, and $M \in \mathbb{N}$ is the number of features. Eq. (3) and Eq. (4) are used to detect symmetry and separability, respectively.

$$F(x_0, x_1, \ldots, x_M) = F_{NN}(x_0 + \alpha, x_1 - \alpha, \ldots, x_M) \quad (3)$$
$$F(\boldsymbol{x_A}, \boldsymbol{x_B}) = F_{NN}(\boldsymbol{x_A}, \boldsymbol{C_B}) + F(\boldsymbol{C_A}, \boldsymbol{x_B}) - F(\boldsymbol{C_A}, \boldsymbol{C_B}) \quad (4)$$

Where $F_{NN}(\boldsymbol{x}) \in \mathbb{R}$ is a NN approximation of $F(\boldsymbol{x})$, and C_i, α are constants. In this way, data points that are not included in the teacher data are complemented by the approximation by NN. In addition, by changing the operator of the detection formula, symmetry and separability corresponding to the four arithmetic operations can be detected. For example, when symmetry of addition is detected, the variables are compressed by re-creating the features of $x_{01} = x_0 + x_1$. When separability is detected, it is possible to split the data as $g(\boldsymbol{x_A}) = F_{NN}(\boldsymbol{x_A}, \boldsymbol{C_B})$ and $h(\boldsymbol{x_B}) = F_{NN}(\boldsymbol{C_A}, \boldsymbol{x_B})$. The discovery of these laws and data processing facilitates formula discovery using genetic programming.

2.2. Genetic Programming

Genetic programming (GP) is a type of symbolic regression using genetic algorithms (GA). GA is a search method that mimics biological evolution, in which a randomly generated initial population is subjected to selection and evolutionary processes to discover better individuals through successive generations. A feature of GP is to represent an equation with a tree structure, as shown in Fig. 2.

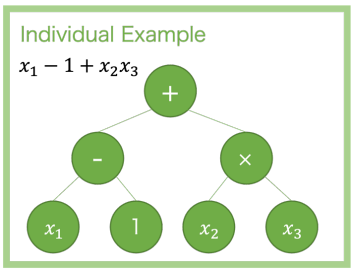
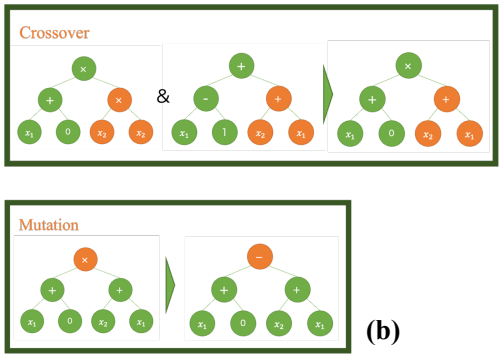


Fig. 2. GP tree example

Each node of the tree structure has the role of an operator or variable and is treated as a gene in the algorithm. The loss of the assignment of teacher data to an individual is the degree of adaptation to the environment, and the lower the loss, the more likely the individual is to remain in the next generation. The remaining individuals undergo an evolutionary process through mutation and crossover to produce the next generation. This process of selection and evolution is repeated to find a better equation.

There are two major types of evolution, as shown in Fig. 3. The first is crossover, in which two selected individuals exchange trees below a certain node. The second, mutation, changes a selected node of one individual to another. The algorithm stops when an individual with losses below a threshold is produced or when a defined generation is reached.



(a)

(b)

Fig. 3. Evolutions. (a) Cross over. (b) Mutation.

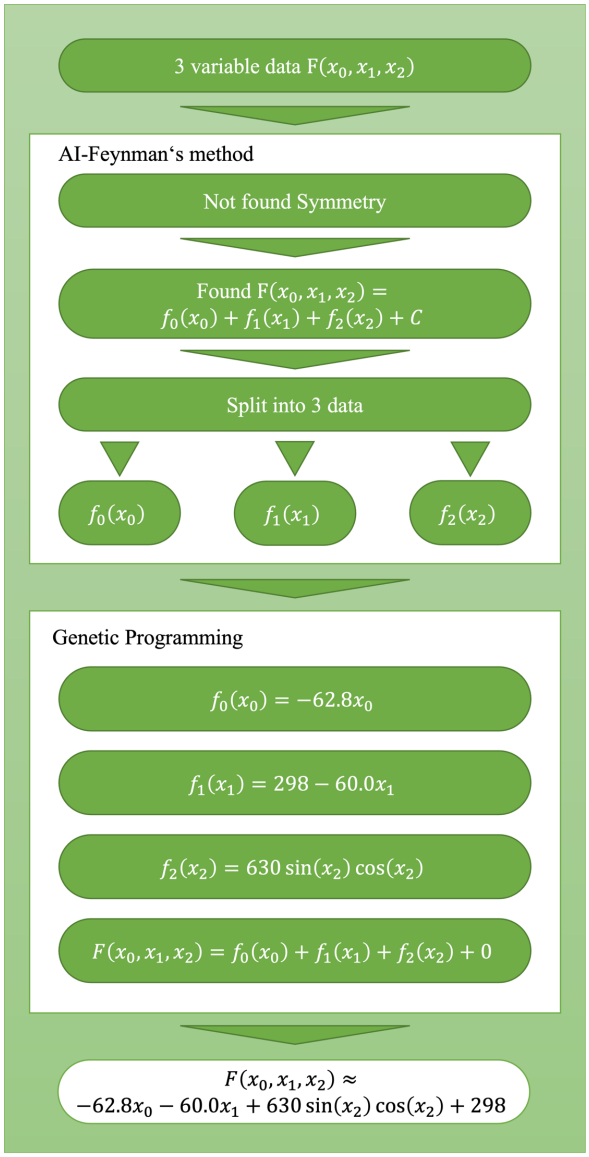


Fig. 4. Flow of Eq. (5).

3. Result example

Fig. 4 shows the flow for discovering an equation from the observed data points. As the example of equation-discovery using the proposed method, we here consider the input-output data pairs produced by the following equation:

$$F(x_0, x_1, x_2) = -63.5x_0 - 62.9x_1 + 629\cos^2\left(x_2 - \frac{\pi}{4}\right). \quad (5)$$

First, the dataset is divided into three datasets after detecting Separability of the addition method by using the AI-Feynman. Then, GP is used to find the equations f_0, f_1, and f_2 for each of the split datasets. GP is also used to find the relationship connecting F and each f_i. Finally, the relationship among the equations is analyzed, and we can obtain the following equation:

$$\tilde{F}(x_0, x_1, x_2) = -62.8x_0 - 60.0x_1 + 630\sin(x_2)\cos(x_2) + 298, \quad (6)$$

which has a similar form with the original equation (Eq. (5)). Figure 5 shows a comparison of $F(x_0, x_1, x_2)$ shown in Eq. (5) and $\tilde{F}(x_0, x_1, x_2)$ shown in Eq. (6). We can see the similar outputs.

The other inferred equations and the Normalized Mean Squared Error (NMSE) for the training data are shown in Table 1. The original equations can be inferred with relatively low NMSEs.

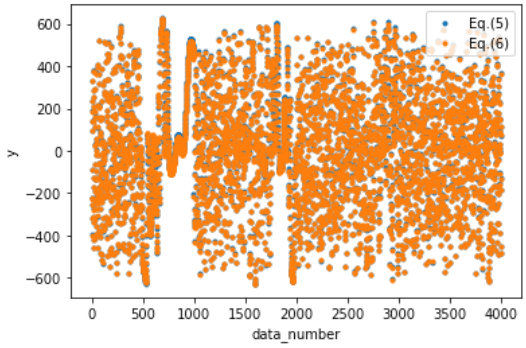


Fig. 5. Comparison of Eq. (5) and Eq. (6).

Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Numbers JP20H04255, JP22H01426, MEXT KAKENHI Grant Number JP22H05198, and JST PRESTO (Grant Number, JPMJPR19M4).

References

[1] https://www.datarobot.com/platform/eureqa-models/

[2] https://turingbotsoftware.com/

[3] Hernandez, A., Balasubramanian, A., Yuan, F., Mason, S. A., & Mueller, T. (2019). Fast, accurate, and transferable many-body interatomic potentials by symbolic regression. *npj Computational Materials*, *5*(1), 1-11.

[4] Udrescu, S. M., & Tegmark, M. (2020). AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, *6*(16), eaay2631.

[5] Udrescu, S. M., Tan, A., Feng, J., Neto, O., Wu, T., & Tegmark, M. (2020). AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Advances in Neural Information Processing Systems*, *33*, 4860-4871.

Table. 1. Comparison of Original equation and Inferred equation.

Original equation	Inferred equation	NMSE
$3(x_0 + x_1) + 2x_2x_3$	$2.73(x_0 + x_1) + 2x_2x_3 - 0.342$	0.00691
$\frac{6.67x_0x_1}{(x_2 - x_3)^2}$	$\frac{7.57x_0x_1}{(x_2 - x_3)^2}$	0.253
$x_0^2 + x_1^2$	$x_0^2 + x_1^2$	0
$\log(2x_0) + 3x_1$	$\log(2x_0) + 2x_1 - 1.18$	0.147
$2\tanh(x_0) - 3\cos(x_1)$	$2\tanh(x_0) + \sin(\tanh(x_0)) - 2.89\cos(x_1) - 0.05$	0.00318
$-10(x_0 - x_1)$	$-10(x_0 - x_1) + 0.0524$	1.52e-6
$-63.5x_0 - 62.9x_1 + 629\cos^2\left(x_2 - \frac{\pi}{4}\right)$	$-62.8x_0 - 60.0x_1 + 630\sin(x_2)\cos(x_2) + 298$	0.000331